Analyzing the data using Live Pie Charts with D3.js

Dinesh Kumar Paladhi

UIN:01019388

Dpala001@odu.edu

Bhavani Manthena

UIN:01033879

Bmant001@odu.edu

Yasaswini Surapaneni

UIN:01052996

Ysura001@odu.edu

**ABSTRACT**

In this paper we will make you walk through how charts can be implemented using D3.js, especially Pie charts. The data that we are using for generating the Pie charts is live data, so the changes can be made to the data there and see its effects in the pie chart.

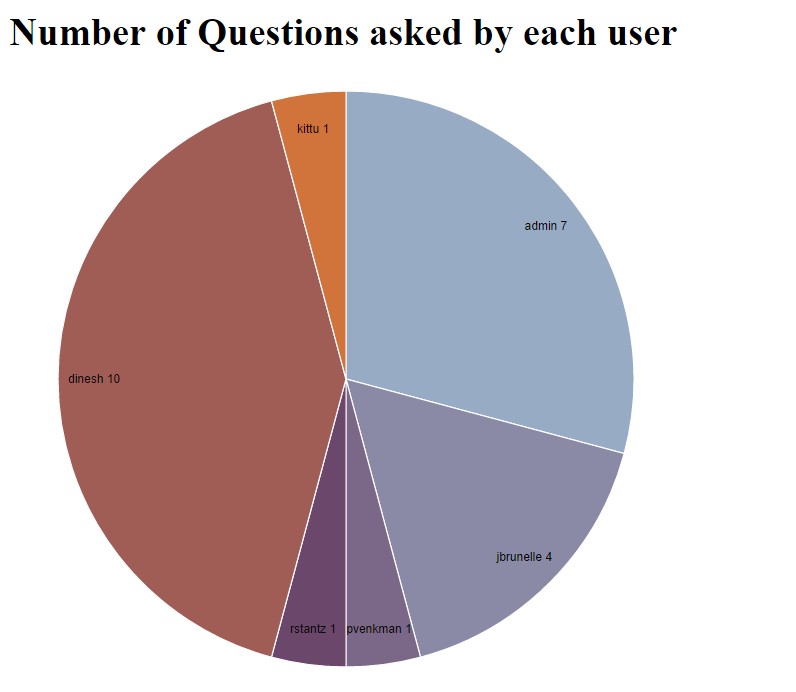
**Keywords**

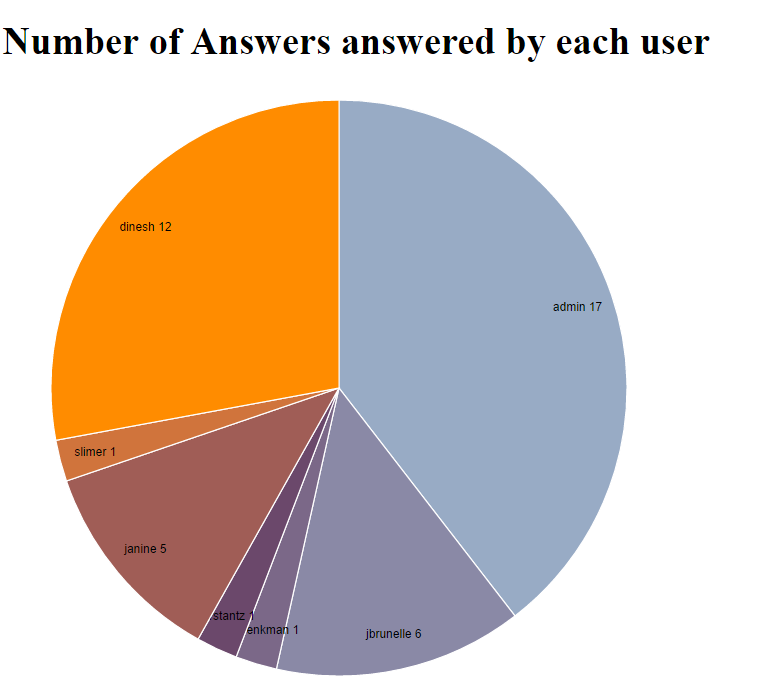
D3.js

# INTRODUCTION

Analysis is a key part for any product after deployment. Analyzing our results will give us the power to solve more complex problems and also improve our results. We can know the trending views of our results by which our performance can be optimized. Here, we have done this analysis part in our project using Pie chart generate by D3.js.

# VIEWS

View of the Pie charts are shown here. The main idea here is to represent the number of questions asked by each user and the number of answers answered by each user. This shows us which user is highly active and which user is not at all participating in the interaction. We can also know if there are some users who only ask questions but does not give any answers. These kind of users are given less score. Pie chart 1 shows us the number of Questions asked by each user and this is dynamic data. The chart reads data dynamically and gives us up to date information. In the view we can see the share of each user compared to other user and also the number of questions asked by every user.



In the above Pie chart, we can see the number of answers answered by each user. This is also dynamic data similar to the other chart. We can see that admin user is the most active user giving higher number of answers to questions on Anskerskart. Next to admin stands dinesh which 12 answers given. These two charts are shown on a single page.

# IMPLEMENTATION

## Converting sql data to csv format

We started the implementation of this idea by first converting the sql data into csv format. For this we used “fputcsv” command to store the sql data into csv file. The data we get from sql query is in the form of an array. We got each value from the array using while loop and “mysqli\_fetch\_row” command.

## Writing to a file in php

Once we got the csv data, we wrote it to a file and saved it in our directory. So every time when we do changes by asking a question or answering to any question, the queries that are written to generate csv are run and the updated data is written to a file and saved in the dictionary.

## Including D3.js Library

Next step was to work with pie chart using D3.js. First we implemented the pie chart with dummy data. In order to work with D3.js, we need to include its library using “<script src="//d3js.org/d3.v3.min.js"></script>”. This library provides us the complete tools for implementing the Pie chart. The dummy data worked well. Understanding D3.js code was a tough task to get through. Lot of errors popped up and it tool lot of time for us to cope up with the dynamic nature of the code. D3.js gives us a lot of flexibility to visualize our data.

## Feeding csv to D3.js Code

Once the D3.js code for the pie charts is ready, we moved on to implement it with changing data. The D3.js Pie code that we wrote takes only csv data as input. So we first made sure that the csv file is correctly written. Then feeding of csv file to D3.js code is done in the following way:

d3.csv("piechart", type, function(error, data1) {

if (error) throw error;

-----so on ----

}

My csv file has two columns user and question for 1 file and user and answer for other file. These columns can be accessed using “d.data.user” , “ d.data.answer”,etc. “piechart” in the above code is the name of the csv file.

## Embedding two charts in page

The next bigger task for us was to embed both the charts in a single page. We also need to trigger both the charts from our AnswersKart Project in order to get updated and dynamic visualization. This was a tough task to do so, We thought of the Html Frames concept and implemented that here. It can be explained clearly with the following code:

frameset cols="50%,50%">

<frame src="quesvis.php">

<frame src="ansvis.php">

</frameset>

This code divides the page into two equal halfs and each half is again like a single separate page. The two pages are included using <frame src> tag. So, We implemented to charts in two separate php pages and clubbed both of them into a single page. These results from this piechart can be alarming if we keenly observe them periodically.

# ISSUES

It was a great experience for us to work with D3.js. It is very flexible and can give us extraordinary visualization. It can be a revolutionary change in the world of visualization if used perfectly.

We have faced a lot of issues in implementing this project. Bascially, it took time for us to understand the D3.js syntax and code. Including the csv file with D3.js code was a tough task and also accessing the included csv data was tougher. We got errors such as invalid file because first time the csv file was not correctly written. We also faced issues in setting the Pie chart dimensions correctly.

We had issues in including both the charts in a single file so we used html frameset.

# REFERENCES

1. Mike Bostock’s Blocks*.* [*https://bl.ocks.org/mbostock/3887235*](https://bl.ocks.org/mbostock/3887235)
2. Writing into a file in php [*http://stackoverflow.com/questions/1768894/how-to-write-into-a-file-in-php*](http://stackoverflow.com/questions/1768894/how-to-write-into-a-file-in-php)
3. Convert array into csv using fputcsv[*http://stackoverflow.com/questions/14809133/php-how-to-convert-array-into-csv-using-fputcsv-function*](http://stackoverflow.com/questions/14809133/php-how-to-convert-array-into-csv-using-fputcsv-function)